Using Automatically Learnt Verb Selectional Preferences for Classification of Biomedical Terms

Irena Spasic,[a] and Sophia Ananiadou[b,c]

[a] Department of Chemistry, UMIST, Faraday Building, PO Box 88, Sackville Street, Manchester M60 1QD, United Kingdom

[b] Computer Science, Multimedia & Telecommunications, School of Computing, Science and Engineering, University of Salford, Salford M5 4WT, United Kingdom

[c] National Centre for Text Mining (NaCTeM), Manchester, United Kingdom

Corresponding author: **Irena Spasic**

Address: Department of Chemistry
UMIST, Faraday Building
PO Box 88
Sackville Street
Manchester M60 1QD
United Kingdom

e-mail: I.Spasic@umist.ac.uk

phone: +44 161 200 4414

fax: +44 161 200 4556

Abstract

In this paper, we present an approach to term classification based on verb selectional patterns (VSPs), where such a pattern is defined as a set of semantic classes that could be used in combination with a given domain-specific verb. VSPs have been automatically learnt based on the information found in a corpus and an ontology in the biomedical domain. Prior to the learning phase, the corpus is terminologically processed: term recognition is performed by both looking up the dictionary of terms listed in the ontology and applying the C/NC-value method for on-the-fly term extraction. Subsequently, domain-specific verbs are automatically identified in the corpus based on the frequency of occurrence and the frequency of their co-occurrence with terms. VSPs are then learnt automatically for these verbs. Two machine learning approaches are presented. The first approach has been implemented as an iterative generalisation procedure based on a partial order relation induced by the domain-specific ontology. The second approach exploits the idea of genetic algorithms. Once the VSPs are acquired, they can be used to classify newly recognised terms co-occurring with domain-specific verbs. Given a term, the most frequently co-occurring domain-specific verb is selected. Its VSP is used to constrain the search space by focusing on potential classes of the given term. A nearest-neighbour approach is then applied to select a class from the constrained space of candidate classes. The most similar candidate class is predicted for the given term. The similarity measure used for this purpose combines contextual, lexical and syntactic properties of terms.

Keywords: term recognition, term classification, ontologies, machine learning, genetic algorithms, similarity measures, corpus processing

1 Introduction

Breakthrough technologies often give rise to large production of data in some scientific disciplines, where the production rate can even exceed that of their analysis. Such a phenomenon is currently most evident in the family of bio-sciences, where the new advances in biotechnology have enabled scientists to experiment at the gene level. New discoveries result in new concepts and their relations being identified, which are described in scientific papers (most often electronically available) with the intention of sharing the new discoveries with the scientific community. However, the corresponding expansion of the bio-literature makes it increasingly difficult for domain experts to access the right information at the right time. For example, the MEDLINE database [1] currently contains approximately 12 million references to journal articles, expanding for more than 10,000 references weekly. For example, over 460,000 references were added in 2003. The sheer volume of the bio-literature could eventually result in a paradox of having too much information causing similar effects as having too little information. In extreme cases locating the information of interest could consume more time than repeating the actual experiments. This would lead to unnecessary repetitive findings instead of progressively founding new discoveries on the existing results.

Clearly, in order for biomedical experts to experience the full benefits of electronically accessible literature, the domain-specific knowledge needs to be organised so as to provide effective means of communication within the domain. The efficient communication should be supported not only between experts, but between computer systems or between experts and computer systems as well. *Ontologies* describe domain-specific knowledge and facilitate information exchange, and as such are a particularly suitable solution for tackling the problem of information overload in biomedicine. Ontologies are scientific models that support clear communication between users, and, on the other hand, store information in a structured form, thus providing support for automatic processing [2]. In particular, ontologies in the biomedical domain model biomedical concepts by providing a semantic framework for bioinformatics tasks such as systematic annotation of domain-specific data or querying heterogeneous language resources [3].

Ontologies can be coupled with natural language processing (NLP) techniques (such as information retrieval, information extraction, etc.) to facilitate the navigation through huge volumes of scientific documents. However, ontologies organise concepts and in order for them to be utilised effectively in NLP applications, they need to link concepts to terms as their linguistic realisations [4]. Moreover, terminological information stored in the ontologies in a structured form needs to be *up-to-date* in order to support access to up-to-date information described in new documents. Particularly, the newly published articles in biomedicine are swamped by newly coined terms denoting newly identified or created compounds, genes, drugs, reactions, etc. The biomedical knowledge sources (such as UMLS [5]) need to integrate and disseminate the new information efficiently in order to allow the experts easy access to new discoveries. Due to an enormous number of terms and the complex structure of biomedical terminologies (e.g. UMLS currently contains over one million concepts named by 2.8 million terms, organised into a hierarchy of 135 classes and interconnected by 54 different relations), manual update approaches are inevitably inflicted by inefficiency and inconsistency. This requires means of automatic extraction of terms, their properties and mutual relations from a corpus of domain-specific documents. At the very least, term recognition, clustering and classification need to be implemented as support for efficient term management and their incorporation into the existing knowledge repositories.

Automatic term recognition (ATR) methods identify isolated pieces of domain-specific knowledge (i.e. concepts represented by terms) and as such are not sufficient when it comes to organising newly acquired knowledge. Concepts are natively assorted into groups and a well-formed model of the domain, represented by an ontology, needs to reflect this property consistently. A dynamic domain model should be able to efficiently adapt to the advent of new terms. In other words, newly extracted terms need to be incorporated into an existing ontology by associating them with one another and with already established terms preferably in an automated manner. *Term clustering* (the process of linking semantically similar terms together) can be used to detect domain-specific associations between terms, while *term classification* (the process of assigning terms to classes from a pre-defined classification scheme) can be used to place new terms into an existing structure through semantic typing of newly recognised terms.

In this paper, we compare some of the term classification approaches and suggest two new approaches to this problem. The paper is organised as follows. In Section 2 we provide an overview of term classification approaches. Section 3 introduces the basic idea of our approach, which uses domain-specific verbs as contextual features for classification of co-occurring terms. Section 4 gives details on the pre-learning phase in which the relevant terminological information is extracted from the corpus. Section 5 describes the learning phase in which domain-specific verb selectional preferences are acquired. Further, Section 6 overviews the main characteristics of a term similarity measure that can be used in a nearest-neighbour approach to term classification. Section 7 describes the classification phase, which combines verb selectional preferences and a nearest-neighbour method. Finally, in Section 8 we describe the evaluation strategy and provide the results, after which we conclude the paper.

2 Related Work

Term classification (as a specific task of text mining) in the biomedical domain is by no means straightforward to implement, because the naming conventions usually do not systematically reflect particular functional properties or relations between biomedical concepts. For example, there is no exact consensus on what constitutes a term even when it is restricted to e.g. proteins and genes [6], although the naming conventions do exist for such concepts [7]. Recently, much of the attention in the biomedical field has been given to identification of protein-protein interactions [8]. In order to extract information about these interactions, the first step is to extract terms belonging to the class of proteins. Therefore, much of the work has aimed at recognition of *specific* classes of terms. The process of identifying terms belonging to the set of prespecified classes and their mapping to the corresponding classes is called *named entity recognition* (NER). On the other side, term classification does not include term recognition. More precisely, term classification is performed after term recognition, while classification is performed as part of named entity recognition.

With that respect, the NER problem can be considered more complex, because it involves two different tasks. However, NER is restricted to specific classes of terms (usually a small number of classes),

which allows such methods to focus on a small set of specific features that characterise such terms. On the other hand, term classification has a difficult task of dealing with terms in general, where terms are not restricted to specific classes of concepts, but instead correspond to arbitrary concepts in the domain. This fact makes it more difficult to develop good-performing term classification methods, because the usual approaches taken in the NER methods, which manually identify typical features of terms from the observed classes, are simply not feasible when a broad set of classes is considered (e.g. UMLS [5] consists of 135 classes). Most of the methods developed for classification of biomedical terms are in fact NER methods. There are very few general term classifications methods. In this section we review both term classification and NER. In the latter case, we focus on the classification aspects of the NER methods, rather than term recognition.

Fukuda et al. [9] developed one of the first NER methods for the recognition of protein names. In their approach, Fukuda et al. explored orthographic and lexical features of protein names. The core features correspond to the distinctive orthographic characteristics of protein names such as capital letters, digits, special characters (e.g. `p54 SAP kinase`). In addition, they define function terms (or f-terms) as keywords (e.g. `protein`, `receptor`, etc.) describing the protein function. Function terms are used to recognise multi-word protein names (e.g. `Ras GTPase-activating protein (GAP)`, `EGF receptor`). These features are used in a rule-based approach to detect protein names in a free text. Focusing on specific types of terms and their features resulted in high precision (94.70%) and recall (98.84%).

A series of other NER methods have been implemented following the original idea of Fukuda et al. For example, Narayanaswamy et al. [6] extended Fukuda's idea to six classes of biomedical entities: gene or protein, gene or protein part, chemical, chemical part, source and general (i.e. all other types of biomedical entities). Analogously to Fukuda et al., Narayanaswamy et al. used the idea of core features and function terms dividing specific features between the classes. For example, chemical root forms based on the IUPAC [10] conventions in naming chemicals are used as the core features for the class of chemicals

(e.g. `–ic acid` as in `suberoylanilide hydroxamic acid`). Similarly, function terms are assigned to classes of entities which they denote. In addition, Narayanaswamy et al. examine *contextual features* (called help words or h-terms) when there are no inner features that can be used for classification. For example, `expression`, `homolog`, `recombinant` are context words associated with proteins.

In order to overcome the problem of manually defining classification rules, many classification approaches resort to machine learning (ML) techniques. These techniques are most often statistically based (e.g. hidden Markov models, naive Bayesian learning, etc.). Other techniques include decision trees, inductive rule learning, support-vector machines, genetic algorithms, etc. Currently, the ML term classification systems exploit little or no domain-specific knowledge for guided learning. Usually, general-purpose ML algorithms are used with shallow representation of text [11]. For instance, Stapley et al. [12] used a support-vector machine (SVM) approach with a non-structured representation of text in order to classify gene names with respect to their sub-cellular location (11 location classes were used). Gene names have been recognised in a dictionary-based approach. Each gene name is represented as a vector of its contextual features, defined as words co-occurring in the same abstract. An SVM approach has been applied to learn the textual features that correspond to particular sub-cellular locations from the labelled training data. This method achieved 60.36% for the macro-averaged F-measure.

Recently, there have been a number of other applications of SVMs for classification of biomedical terms. These approaches differ from that of Stapley et al. with respect to the features used. Mostly, the features used resemble those proposed by Fukuda et al. [9]. In addition, such methods are used for NER rather than term classification (as is the case in [12]). For example, Kazama et al. [13] used an SVM for the recognition of six types of biomedical entities (e.g. `protein`, `DNA`, etc.). Various types of features are considered including lexical, morphologic and orthographic properties. For example, there is a feature for each word in a given vocabulary, POS that can be assigned by the tagger, prefix and suffix in a fixed list, substring from a fixed list, and class that can be assigned to an individual word. All these features are considered for both terms and their contexts. The F-measure of the corresponding classification method reached 54.4%. Lee et al. [14] also used an SVM approach for NER. They relied on the same features as

Kazama et al. However, they explicitly divided the recognition and classification tasks of NER. Each of these tasks is dealt with by separate SVMs. The recognition of named entities is performed by an SVM analogous to that of Kazama et al. A separate SVM is developed for each class using class-specific features similar to those proposed in [9] and contextual features such as the presence of a certain noun/verb in the left/right context. The final classification results are obtained through a voting procedure, where each class-specific SVM is allowed to vote. The classification performed in this manner achieved 66.50% for F-measure. In another SVM-based approach to NER, Takeuchi and Collier [15] used similar features defined for ten biomedical classes including protein, DNA, RNA and seven types of biological sources. Unlike other previously described SVM approaches, which used linear classification function (kernel), Takeuchi and Collier used a polynomial kernel function and achieved 74% for F-measure.

The current popularity of SVMs for classification of biomedical terms is due to their simplicity and robustness with respect to high-dimensional feature spaces. However, they need large amounts of training data in order to perform well. They also underperform for minority classes due to the sparse data problem. In addition, they may be impractical when the classification is performed against a large number of classes. Alternatively, probabilistic methods such as naive Bayes classification have been used widely. For example, Nobata et al. [16] implemented such method for term classification. The goal of naive Bayes classification is to maximise the conditional probability of a given term being assigned to a specific class based on the features used to represent a term. This probability can be estimated as the product of the class probability and the conditional probabilities of features given the class, based on the hypothesis that these features are independent. The features used by Nobata et al. include individual words used to build terms. The independence assumption implies that each word occurrence is independent of its context and position in the text. This restriction may seem to strong, but it has been successfully employed in text classification [17]. A shortcoming of the specific choice of features in this approach is its inapplicability to unknown terms, namely the ones consisting only of "unknown" words (the ones for which no classification probabilities have been pre-determined).

While Nobata et al. statistically processed information found inside terms, Hatzivassiloglou et al. [18] applied the same approach to information found outside, i.e. the contexts in which term occurrences were found. The context is represented as a bag of words. The features used to represent an individual context word include its stem (in order to neutralise inflectional and derivational variations), part-of-speech and position relative to the term being classified. F-measure obtained by this approach dropped from 84% to 73% when going from two to three classes. The precision is expected to drop even further in general classification with a high number of classes.

Collier et al. [19] also applied statistical techniques to contextual information. Their classification method is theoretically founded on the hidden Markov models (HMMs), which implement stochastic finite state automata and have been widely used in NLP for POS tagging. Collier et al. based a HMM for term classification on *n*-grams assuming that term's class may be induced from previous $n - 1$ lexical items (e.g. terms) and their classes. In their implementation, a word was defined as an ordered pair consisting of a word's surface form and its features. They relied on orthographic features as, in biomedicine, these often provide hints regarding a class of a specific term. The F-measure achieved for twelve classes of genes and their products was 73%.

3 Term Classification Using Term-Verb Co-occurrence Pairs

In the previous section, we discussed some of the state-of-the-art approaches to classification of biomedical terms. There are two basic choices that characterise each of these approaches, which concern *features* and a *method* to be used. Features can be divided into *internal* and *contextual* features with respect to their position relative to the term being classified. There is also a choice of the types of features to be used, e.g. *orthographic*, *morpho-syntactic*, etc. We chose to rely on contextual features rather than internal ones, because the naming conventions in biomedicine usually do not systematically reflect particular functional properties or relations between concepts. In addition, we wanted to avoid manual identification of specific features. Further, the methods used for classification typically belong either to rule-based ap-

proaches or machine learning family. We opted for machine learning rather than a rule-based approach again in order to minimise the need for handcrafted knowledge.

This leads us to the specific choice of features and methods to be used for classification. The basic hypothesis governing our choice of features is that the meaning of linguistic elements (e.g. represented by their semantic classes) is related to the restrictions according to which these elements may be combined [20]. In other words, this distributional hypothesis states that specific linguistic relations apply to semantically similar words. For example, only words from restricted semantic classes can appear in certain predicate-argument structure. In particular, nouns can be used as a subject or object of a restricted set of verbs. It follows that each noun can be characterised by the set of co-occurring verbs. For example, Hindle [21] based his work on noun classification on this hypothesis. Similarly, in the biomedical sublanguage, [18] used this hypothesis to automatically discover the facts such as the one that specifies that `proteins activate genes`, and not vice versa. The patterns describing `activation` relation were matched against a text to extract facts that "`x activates y`". Terms found to be in this relation were mapped to their semantic classes and statistically processed, which resulted in the fact that `x` was a `protein` in the majority of cases, while `y` was a `gene`.

In this work, we modify (or specialise) the distributional hypothesis in terms of a specific sublanguage: each term can be characterised by the set of co-occurring verbs, or, conversely, each domain-specific verb (DSV) can be used in combination of a restricted set of semantic classes. We will, therefore, use DSVs as features for classification of the co-occurring terms. When we know the classes that can be used to complement a fixed DSV, then the above assumption can be used to infer possible classes for the co-occurring terms. The necessary step in this approach is to identify DSVs and acquire selectional preferences for each identified verb.

DSVs can be acquired automatically by using statistical information including frequency of occurrence combined with frequency of co-occurrence with terms. Namely, similarly to terms denoting domain-specific concepts, DSVs are used to linguistically describe relations specific in the domain. They

will naturally have a high frequency of occurrence. In addition, as they describe relations between concepts, they will consequently co-occur with terms denoting the concepts involved.

Further, once DSVs are identified, we need to acquire their selectional preferences in the form of classes whose terms can be meaningfully used with them. Given a DSV, the simplest approach to acquiring "compatible" semantic classes would be to map each co-occurring term to its classes and use the frequency information obtained for these classes (e.g. [22]). In this work we extend this approach by further generalising these classes by climbing the hierarchy of classes. In an alternative ML approach based on genetic algorithms, we learn an "optimal" set of classes to be combined with a given verb, optimal in the sense that it minimises the number of false positives and false negatives. Once the selectional preferences are available, this information is used to classify newly recognised terms used in combination with the DSV. A term is linked to a specific class by estimating the similarity between the term and the classes typically selected by the co-occurring verb. A term similarity measure that combines lexical, syntactic and contextual properties of terms [23, 24] is used for this purpose.

4 Corpus Processing and Analysis

We used a domain-specific corpus consisting of 2072 abstracts on nuclear receptors retrieved from the MEDLINE database [1]. Each abstract consists of a single title and a number of sentences. The total number of sentences in the corpus (not counting the titles) is 19449, while the total number of word tokens is 558556.

The linguistic and domain-specific information has been annotated in the corpus by using XML. This information has been obtained automatically and no manual intervention has been used. First, each word has been tagged[1] with the information about its lexical class, morphological description and lemmatised form. The corpus was terminologically processed in order to recognise term occurrences. Terms were recognised by the C/NC-value method [25] and an ontology derived from UMLS [5] (namely, a six-level deep subtree of the UMLS ontology whose root corresponds to biochemical substances, which is further

refined into 28 classes) was used to locate classified terms in the corpus. A total of 2757 term and 28935 of their occurrences were recognised by the C/NC-value method in addition to 2609 terms and 29636 of their occurrences recognised from the ontology.

The terminological information was annotated by the LEXIE tool during the shallow syntactic parsing used to identify the syntactic categories of interest (e.g. noun phrases, auxiliary verb phrases, etc.). The annotated corpus was analysed in order to recognise domain-specific verbs. Once both terms and DSVs have been obtained, their co-occurrences are extracted to be used for learning verb selectional preferences. Let us describe how domain-specific information is extracted in more details.

4.1 Term Recognition

First, the corpus is terminologically processed: both terms present in the ontology and the terms recognised automatically are tagged. Terms already classified in the ontology will be later used to learn the classes allowed by the DSVs, while the new terms are yet to be classified based on the learnt classes. New terms are recognised by the *C/NC-value method* [25], which extracts multi-word terms (more than 85% of domain-specific terms are multi-word terms [26]). This method recognises terms by combining linguistic knowledge and statistical analysis. Linguistic knowledge is used to propose term candidates through general term formation patterns. The proposed candidates are then statistically processed. Each term candidate t is quantified by its termhood, denoted as *C-value*(t), calculated as a combination of the term's numerical characteristics: length $|t|$ as the number of words, absolute frequency $f(t)$ and two types of frequencies relative to the set $S(t)$ of candidate terms containing a nested candidate term t (frequency of occurrence nested inside other candidate terms and the number of different term candidates containing a nested candidate term):

$$C-value(t) = \begin{cases} \ln |t| \cdot f(t) & , \text{ if } S(t) = \varnothing \\ \ln |t| \cdot (f(t) - \frac{1}{|S(t)|} \sum_{s \in S(t)} f(s)) & , \text{ if } S(t) \neq \varnothing \end{cases}$$

[1] The corpus was annotated by the LEXIE tool during the BioPATH project. LEXIE is an NLP software package, which uses a HMM-based POS tagger and performs shallow parsing based on finite state automata. It was developed at the Text Mining Group, LION BioScience (Heidelberg, Germany) as part of the EUREKA BioPATH project.

Obviously, the higher the frequency of a candidate term the greater its termhood. The same holds for its length. On the other side, the more frequently the candidate term is nested inside other term candidates, the more its termhood is reduced, because it is assumed to be only a building block for other terms rather than an independent term itself. However, this reduction decreases with the increase in the number of different host candidate terms as it is hypothesised that the candidate term is more independent if the set of its host terms is more versatile.

Term distribution in top-ranked candidate terms is further improved by taking into account their contexts, because terms tend to co-occur with certain lexical categories [4]. The relevant context words, including nouns, verbs and adjectives, are extracted from the corpus and assigned weights based on how frequently they co-occur with top-ranked term candidates. Subsequently, context factors are assigned to the candidate terms according to their co-occurrence with top-ranked context words. Finally, new termhood estimations (NC-values) are calculated as a linear combination of the C-values and context factors.

Nenadic et al. [27] modified the C/NC-value method to recognise acronyms as a special type of single-word terms, and, thus, enhanced the recall of the method. On the other hand, the modified version incorporates the unification of term variants into the linguistic part of the method, which also improved the precision, since the statistical analysis proved to be more reliable when performed over classes of equivalent term variants instead of separate terms.

4.2 Domain-Specific Verb Recognition

We used a simple method to extract domain-specific verbs from the corpus though we are well aware that more sophisticated methods should be used in future experiments. First, 55576 verb occurrences were extracted and mapped to their infinitive form by using morphological information annotated in the corpus during POS tagging. As a result, these occurrences were mapped to 1008 different verbs, which were ranked according to their frequency of occurrence. Some of the highest ranked verbs were general verbs (e.g. the verbs `be` and `have` occurred 14171 and 2453 times respectively). A manually assembled stoplist was used to filter out such verbs. A frequency threshold was used to eliminate less frequently occur-

ring verbs. A total of 88 verbs occurring more than 100 times were retained. These verbs were re-ranked according to their frequency of co-occurrence with terms. Table 1 provides a list of ten high-ranked verbs, which were later used in the experiments. The selected verbs are considered to be domain-specific. Moreover, these verbs are also corpus-specific.

activate	mediate
bind	modulate
block	regulate
inhibit	repress
interact	stimulate

Table 1. Domain-specific verbs

4.3 Term-Verb Co-occurrence Pairs

We adopted a heuristic approach to extracting terms used as subject, object or object of a preposition for the given DSVs. First, we noted that transitive verbs dominated the given list of DSVs (only `interact` is an intransitive verb). Table 2 provides the most frequent syntactic patterns in which terms are combined with the given verbs. Such patterns typically denote the existence of a certain relationship between two terms. We thus expected for most of the analysed verbs to co-occur with a term in both left and right context. However, it is not obligatory for a term to be a direct neighbour of a given verb. Consider, for example: `SF-1 was recently shown to interact with DAX-1`. We, therefore, did not limit ourselves only to extraction of terms immediately preceding or following a verb. Instead, we used co-occurrence in a text window of a fixed length. In our approach, we extracted the closest terms occurring at most three positions away (without crossing the sentence boundary) from the verb considered, where the positions refer to syntactic chunks annotated during the shallow parsing rather than individual tokens. In the previous example, the following rules were applied:

```
<report-VP>   ::=  [ ( has | have ) been ] <adv> <report> [ that | to ]
<report>      ::=  anticipated | assumed | believed | claimed | concluded | con-
                   firmed | considered | demonstrated | deduced | described | de-
```

termined | discussed | envisaged | envisioned | expected | found | hypothesized | identified | inferred | interpreted | judged | known | noted | observed | predicted | presumed | proposed | proven | proved | purported | postulated | predisposed | recognized | reported | revealed | said | seen | shown | speculated | suggested | supposed | suspected | thought | tended | used

which resulted in the following annotation:

<TERM>SF-1</TERM> <report-VP>was recently shown to</report-VP>

<VERB>interact</VERB> <PREP>with</PREP> <TERM>DAX-1</TERM>.

in which term SF-1 is only two positions away from the verb interact instead of five positions had the above rules not been used.

A small distance between a term and the co-occurring verb is used to indicate that there may be a syntactic relation between them. Namely, if a term co-occurs closely with verb, then we assume that it is its subject, object or object of a preposition. Some of the relations denoted by DSVs are symmetric (e.g. verb interact). In other words, if COUP-TF II interacts with p300, then it is also true that p300 interacts with COUP-TF II. This means that these terms, when used in combination with the given verb, can freely exchange the functions of a subject or an object of the preposition with without affecting the overall meaning. Therefore, in this example, there is no need to analyse the syntactic function of the co-occurring terms, since the classes inferred for the subject of the given verb would be the same for the corresponding object of the preposition. However, there are verbs for which different classes apply to different syntactic relations. For example, let us recall that proteins activate genes and not vice versa. This means that terms belonging to the class of proteins may act as a subject of the verb activate, but not its object, and, conversely, terms from the class of genes can act as an object of the given verb, but not its subject. In our approach, the generalisation of this fact would be that genes and proteins are semantic classes that can be combined with the verb activate. So both classes can be used as potential classes for the terms co-occurring with the given verb.

We use a similarity measure between a term and these classes to resolve such an ambiguity. Namely, given a verb, even a single syntactic function could typically be generalised into *multiple* semantic classes. This means that for the purpose of term classification, *additional processing* would typically be needed even when the syntactic function is resolved. This could be done in a previously described similarity-based approach. Still, to distinguish different syntactic roles may improve the performance by further constraining the set of potential classes (as is the case with the verb `activate`), which is one of the future research topics.

Verb	Pattern	Example
intransitive	<Term> <Verb> with <Term>	`aryl hydrocarbon receptor interacts with estrogen receptor alpha`
transitive, active	<Term> <Verb> <Term>	`C terminal tail inhibits the Pitx2 protein`
transitive, passive	<Term> <Verb> by <Term>	`chain promoters were inhibited by C/EBPbeta isoforms`

Table 2. Patterns combining terms and domain-specific verbs

5 Learning Phase: Acquisition of Selectional Patterns for Domain-Specific Verbs

As we have discussed, domain-specific verbs impose selectional restrictions (or preferences) on the terms that co-occur with them. The goal of the learning phase of our term classification approach is to acquire these preferences automatically from a domain-specific corpus. We define a *verb selectional pattern* (VSP) as a set of semantic classes that could be used in combination with a given DSV. More precisely, a VSP represents a hypothesis about the classes of terms used with the corresponding DSV. We say that a VSP applies to a term if it is a member of at least one class present in the VSP. In this paper, we suggest two possibilities to learn VSPs automatically. In the first approach, we use a simple generalisation procedure based on the hierarchical organisation of the UMLS ontology. Second approach uses a generic algorithm to optimise a VSP so that the number of terms co-occurring with the corresponding DSV to which the VSP applies is maximised, while minimising the number of classes in the VSP that do not apply (or

rarely do) apply to the co-occurring terms. The following two subsections describe two specific implementations for acquiring VSPs from a domain-specific corpus.

5.1 First Learning Method: Class Generalisation

In this approach, we use a simple generalisation procedure in order to generate a VSP for each DSV separately. The generalisation method is based on a partial order relation induced by the domain-specific ontology.[2] We used a subtree of the UMLS ontology whose root corresponds to biochemical substances, which is further refined into 28 classes.

In the initial phase, all terms co-occurring with a given DSV are collected from the corpus (see Section 4.3) and mapped to the classes assigned to them in the ontology. These classes are used as initial generalisation from terms to their classes and to calculate the frequency of co-occurrence of each class with the given verb. The tree number[3] is retrieved from the ontology for each class. All initially obtained classes are sorted in the descending order based on their tree numbers and processed iteratively in that order. This means that the most specific classes (i.e. the ones corresponding to the deepest nodes in the hierarchy) are processed first. Each class is compared pair-wise to all classes following it in the specified order. Let C_1 denote currently processed class and let C_2 be a class it is compared to. If C_2 is a parent of C_1, then C_1 is "generalised" into C_2.[4] This operation means that C_1 is removed and its frequency is added to that of C_2. This process is repeated until all classes are processed. The remaining classes represent a hypothesised VSP. Each class C_i in the VSP has the frequency feature f_i, which aggregates the frequency of co-occurrence with the given verb. The frequency information is used to estimate the class probabilities given a verb, $P(C_i \mid v)$:

[2] The partial order relation is based on the hierarchy of terms/classes: term/class t_1 is in relation with t_2, if there is a path in the ontology from t_2 to t_1. In that case, we say that t_2 is *more general* than t_1.

[3] Given a node in a hierarchy, the tree number records the relative numbers of the nodes in a hierarchy starting from the root node through each node in the path leading to the given node. Based on this information, two nodes can be easily compared not only for the existence of the general-specific relation between them.

[4] Iterative application of this rule may cause overgeneralisation. To prevent this, the classes placed close to the root of the ontology should be either removed from the input set or prevented from substituting less general classes. The depth up to which the classes are to be blocked may be empirically determined. In our approach, we prevented generalisation for the root classes and its direct children.

$$p_i = \frac{f_i}{\sum_{C_j \in VSP} f_j} \qquad (1)$$

5.2 Second Learning Method: A Genetic Algorithm

Before describing the specific properties of our implementation of a genetic algorithm (GA) for learning VSPs, we provide a brief overview of GAs in general. GAs are meta-heuristics incorporating the principles of natural evolution and the idea of "survival of the fittest" [28]. An *individual* encodes a solution as a sequence of genes. In the initial phase of a GA a number of solutions is generated, usually at random. Selection, crossover, mutation, and replacement are applied in this order aiming to gradually improve the quality of the solutions and the possibility of finding a sufficiently good solution. *Selection* is usually defined probabilistically: the better the solution, the higher the probability for that solution to be selected as a parent. Selected individuals are recombined by applying the *crossover* between pairs of individuals. The offspring is expected to combine the good characteristics of their parents, possibly giving way to better solutions. The *mutation* operator introduces diversity into a population by modifying a solution, possibly introducing previously unseen good characteristics into the population. *Fitness function* quantifies the quality of individuals. The ones with the best fitness values *replace* less fit individuals. Once a suitable solution has been found or the number of iterations exceeds some threshold, the iterations of the GA are stopped.

In our approach, each individual corresponds to a VSP. It is represented as a sequence of genes, where each gene encodes whether the class corresponding to its position is a member of the given VSP or not (so-called bit representation). The concrete problem we aim to solve by the GA is to optimise VSPs so as to minimise the number of classes in a VSP while maximising the number of training terms (co-occurring with the given verb) to which a VSP applies (i.e. the number of terms belonging to at least one class from a VSP). In other words, if we treat each class in a VSP as a class predicted for all terms co-occurring with a given verb, then we are basically optimising precision and recall of such classification, where these measures are calculated as follows:

$$P = \frac{A}{A+B} \qquad R = \frac{A}{A+C}$$

where:

- A if the number of true positives (correctly classified positive instances),
- B if the number of false positives (incorrectly classified negative instances) and
- C if the number of false negatives (incorrectly classified positive instances).

The fitness is then calculated by estimating precision and recall of the VSP on the training set (the same as in the first learning method) by using the ontology as a gold standard. Namely, for each individual training term co-occurring with the given verb, we compare the classes in the VSP to those assigned to the given term in the ontology. The fitness of each solution corresponds to the discrepancy between the classes predicted by the VSP and the actual classes. Given a term t from the training set T, let O_t denote a set of classes assigned to that term in the ontology. The fitness f of a VSP, denoted shortly as V, is then calculated as follows:

$$f(V) = \sum_{t \in T} \left(w^R \cdot | O_t \setminus V | + w^P \cdot | V \setminus O_t | \right) \qquad (2)$$

where w^R and w^P are the weights used to model the performance preferences towards the precision and recall.[5] Each time an actual class has not been suggested by the system the solution is "fined" by adding the recall weight w^R to its fitness, and vice versa, each time an incorrect class is suggested the solution is "fined" by adding the precision weight w^P to the fitness. In extreme cases, one of the weights may be set to zero. For example, the precision weight may be discarded in applications where recall is of the utmost importance (e.g. in medical diagnostic applications), and, alternatively, the recall weight may be set to zero where precision is crucial (e.g. automatic ontology update). In general, positive values should be used, and the ratio between the recall and precision weight should reflect the importance of these characteristic in the overall performance of the system. The objective of the GA in this case is to minimise the fitness function, that is – to find a solution with a (near)minimal fitness value.

[5] In our experiments we used equal weights for precision and recall.

The initial population is formed by generating random solutions (VSPs). The crossover between the solutions is uniform: genes at each fixed position are exchanged with 50% probability. Solutions are mutated with 1% probability, and this operation involves a random change of a randomly chosen gene. Fitness is calculated for all solutions in the current generation and all newly generated solutions. The fittest newly generated solutions replace the least fit solutions in the current generation. The process of crossing over the solutions, mutating them and replacing the least fit solutions by fitter new solutions is repeated until a solution with an acceptable fitness is found, or the number of generations reaches a certain threshold.

The presented GA optimises the recall and precision in general, or, more precisely, the microaveraged recall and precision in which all classes are treated equally. It is also possible to have different preferences towards different classes (for example higher recall for identification of toxic substances), in which case the fitness function given in formula (2) needs to be reformulated. For example, the fitness function that incorporates this property can be formulated in the following way:

$$f(V) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(w_j^R \cdot |(O_i \setminus V) \cap \{C_j\}| + w_j^P \cdot |(V \setminus O_i) \cap \{C_j\}| \right)$$

where n is the number of classes, w_j^R and w_j^P are the weights used to model the performance preferences towards the precision and recall for the class C_j. In our experiments, we have used the fitness function described in formula (2).

6 Nearest-Neighbour Approach to Term Classification

A verb complementation pattern typically contains multiple classes. In order to link the newly recognised terms to specific candidate classes, we used a similarity-based approach.. In this section we review the CLS term similarity measure, and adapt it to be used for term to class comparison.

6.1 Term Similarity

Nenadic et al. [29] used the Dice coefficient to compare lexical, syntactic and contextual term features separately, and then combined the calculated values into a hybrid *CLS* (contextual, lexical and syntactic) term similarity measure. Lexical features used in the CLS measure refer to lexical constituents shared by the compared terms. The rationale behind the lexical term similarity involves the following assumptions [30]: (1) Terms sharing a head are likely to be hyponyms of the same term (e.g. `progesterone receptor` and `oestrogen receptor`). (2) A term derived by modifying another term is likely to be its hyponym (e.g. `nuclear receptor` and `orphan nuclear receptor`). The similarity between two terms t_1 and t_2 is measured by combining information on having a common head and/or modifier(s):

$$LS(t_1,t_2) = 2 \cdot \left(w_h \cdot \frac{|H_1 \cap H_2|}{|H_1| + |H_2|} + w_m \cdot \frac{|M_1 \cap M_2|}{|M_1| + |M_2|} \right)$$

where H_1 and H_2 represent the heads of terms t_1 and t_2 respectively, M_1 and M_2 are the sets of the stems of their modifiers, and w_h and w_m ($0 \le w_h, w_m \le 1$, $w_h + w_m = 1$) are the weights giving different preferences toward shared heads and modifiers (we treated heads as three times more important than modifiers). This measure is simple and effective for comparing multi-word terms, but it falls short when it comes to single-word terms (acronyms in particular) and those introduced in an ad-hoc manner.

Hyponymy relation can be inferred not only from terms themselves (by exploring their lexical features), but also from the contexts in which they occur (by relying on their syntactic features) [31, 32]. Nenadic et al. further defined lexico-syntactic patterns that can be used not only to extract hyponymy relation, but general similarity relation (see Table 3 for examples), assuming that terms that share the same syntactic function within the sentence (e.g. object or subject) in combination with other sentence constituents (e.g. verbs or prepositions) are similar. Namely, the parallel usage of terms within the same context more strongly associates the terms involved than other types of co-occurrence. The value of syntactic similarity for terms t_1 and t_2 depends on the type of patterns in which the terms co-occur and the frequency of co-occurrence of the given terms in these patterns, and is calculated as follows:

$$SS(t_1,t_2) = \sum_{i=1}^{n} w_i \cdot SS_i(t_1,t_2)$$

where n is the total number of lexico-syntactic patterns considered, w_i ($0 \le i \le n$) is the weight given to the i-th pattern ($w_1 + ... + w_n = 1$).[6] The similarity between two terms with respect to the i-th pattern, denoted as $SS_i(t_1, t_2)$, is calculated by the following formula:

$$SS_i(t_1,t_2) = \begin{cases} 0 & \text{, if } f_i(t_1) + f_i(t_2) = 0 \\ \dfrac{2 \cdot f_i(t_1,t_2)}{f_i(t_1) + f_i(t_2)} & \text{, otherwise} \end{cases}$$

where $f_i(t_j)$ is the frequency of occurrence of term t_j (j = 1, 2) inside the i-th pattern, while $f_i(t_1, t_2)$ is the frequency of simultaneous occurrence of terms t_1 and t_2 in the i-th pattern.

<Term> [(] <EG> <Term> (,<Term>)* [<CC> <Term>] [)] <Term> (, <Term>)* [,] <CC> other <Term>
<Term> [,] (including | especially) <Term> (,<Term>)* [[,] <CC> <Term>]
both <Term> and <Term>
either <Term> or <Term>
neither <Term> nor <Term>

Table 3: Examples of parallel lexico-syntactic patterns[7]

While the precision of the syntactic measure is expected to be high due to specific nature of patterns used and the semantic similarity that they strongly imply, the recall of this measure may be low, since not all similar terms are bound to appear in parallel structures. In order to provide higher recall a large-size corpus is required. In order to remedy for small-size corpora, other contextual patterns (CPs) in which terms appear are used as additional features for term comparison. CPs represent abstracted term contexts consisting of the syntactic categories and other grammatical and lexical information (e.g. `PREP NP V:stimulate`). They are ranked according to a measure called CP-value (analogue to C-value for ATR

[6] We used equal weights for all patterns.
[7] Non-terminal symbols <EG> and <CC> are described by the following regular expressions respectively: (such as) | like | (e.g.[,]) and (as well as) | (and[/or]) | (or[/and]).

[25]). CPs with high CP-values are usually general patterns, the ones with low CP-values typically are rare patterns, while the middle-ranked CPs represent relevant domain-specific patterns. Therefore, the ones whose CP-value is inside the chosen boundaries are deemed significant for term comparison (20% of the top-ranked CPs and 30% of the bottom-ranked CPs have been discarded). Each term is associated with a set of the CPs with which it occurs, and contextual similarity between terms t_1 and t_2 is then measured by comparing the corresponding sets C_1 and C_2 by the Dice coefficient:

$$CS(t_1, t_2) = \frac{2 | C_1 \cap C_2 |}{| C_1 | + | C_2 |}$$

Finally, the three similarity measures (lexical, syntactic and contextual) are linearly combined into a hybrid similarity measure:

$$CLS(t_1, t_2) = a \cdot LS(t_1, t_2) + b \cdot SS(t_1, t_2) + g \cdot CS(t_1, t_2)$$

The choice of the weights α, β, and γ is not a trivial problem. Spasic et al. [24] chose to fine-tune the above measure automatically rather then experimenting with manually determined weights. They implemented a supervised learning method based on the ontology. The similarity measure used as a gold standard was calculated by applying the Dice coefficient to the features extracted from a domain specific ontology. The hybrid similarity measure was evaluated according to the Euclidean distance:

$$f(a, b, g) = \sum_{\substack{t_i, t_j \in T \\ t_i \neq t_j}} \left(CLS_{abg}(t_i, t_j) - O(t_i, t_j) \right)^2$$

where the set T is the intersection between two sets of terms, one derived from the ontology and the other automatically recognised terms, $CLS_{\alpha\beta\gamma}(t_i, t_j)$ is the hybrid similarity measure calculated for the given weights α, β, and γ, and $O(t_i,t_j)$ is the similarity measure derived from the ontology. The goal was to find a combination of weights that minimises the value of evaluation function. In other words, the deviation from the similarity values derived from the ontology needs to be minimised. For this purpose, a genetic algorithm was used. As a result, the following values have been obtained: $\alpha = 0.72$, $\beta = 0.11$ and $\gamma = 0.17$.

The reason for the syntactic similarity to be assigned the lowest weight is due to the sparsity of data. As mentioned earlier, this measure requires a large-size corpus in order to provide a reliable recall.

6.2 Term-to-Class Similarity

The CLS similarity measure applies to pairs of terms. However, in case of multiple choices provided by the VSPs, we need to compare terms to classes. In order to do so, we use the similarity between the given term and the terms belonging to the suggested classes. The selection of terms to be compared is another issue. For each class, we have used k (15 in our approach) randomly chosen terms that occur in the corpus as class "representatives". More formally, if c is a class, $e_1, e_2, ..., e_k$ are terms representing the class, and t is a term, then the similarity between the term t and the class c is calculated in the following way:

$$Ex(t,c) = \max_{i \in \{1,...,k\}} \frac{CLS(t,e_i)}{\sqrt{\sum_{j=1}^{k} CLS^2(t,e_j)}} \qquad (3)$$

This example-based similarity measure maximises the value of the CLS measure between the term and the instances representing the class. In addition, the values of the CLS measure are mapped into the interval (0,1) by performing vector normalisation in order to make them comparable to the class probability estimations.

7 Classification Phase

Let us now describe the classification procedure in more detail. Let $V = \{v_1, v_2, ..., v_n\}$ be a set of automatically identified domain-specific verbs. During the phase of learning the VSPs, each of these verbs is associated with a set of classes it typically co-occurs. Let $VSP_i = \{c_{i,1}, ..., c_{i,m_i}\}$ denote a set of classes assigned automatically to the verb v_i ($1 \leq i \leq n$) by a learning algorithm based on the information found in the corpus and the ontology. When classifying a specific term, its frequency of co-occurrence with selected DSVs is used to chose the verb (i.e. its VSP) based on which the term will be classified. Precisely, the verb the given term most frequently co-occurs with is chosen, as it is believed to be the most indica-

tive one for the classification purpose. The actual classification procedure varies slightly depending on the method used for learning the VSPs.

In the first case, that is – when the VSPs are obtained by the first learning algorithm, given the term t and the verb v_i it most frequently co-occurs with, a score is calculated for each class $c_{i,j}$ from the set C_i according to the following formula:

$$C(t, c_{i,j}) = a \cdot p_{i,j} + (1-a) \cdot Ex(t, c_{i,j}) \quad (4)$$

where the factors $p_{i,j}$ and $Ex(t, c_{i,j})$ are calculated according to formulas (1) and (3), while a ($0 \le a \le 1$) is a parameter that balances the impact of the class probabilities and the similarity measure.[8] A class with the highest $C(t, c_{i,j})$ score is used to classify the term t. Alternatively, multiple classes may be suggested by setting a threshold for $C(t, c_{i,j})$.

In the second case, that is – when the VSPs are obtained by the second learning algorithm, given the term t and the verb v_i it most frequently co-occurs with, a score is calculated for each class $c_{i,j}$ from the set C_i according to the following formula:

$$C(t, c_{i,j}) = Ex(t, c_{i,j}) \quad (5)$$

The probability factor is apparently missing in formula (5) compared to formula (4). However, the probability factor is implicitly encoded into the VSP itself. Namely, recall that the second approach optimises the VSPs so as to optimise precision and recall (assuming that each class from the VSP is predicted) given a training set of terms. The more frequently a certain class co-occurs with the given verb the more likely it will be present in the VSP.

Note that terms can be compared directly to all classes in the classification scheme in order to perform the nearest neighbour classification directly instead of using VSPs as mediators. However, the class pruning approach has been adopted in order to enhance the computational efficiency of the classification process itself. Namely, VSPs are used to constrain the search space aiming to reduce the number of classes a

[8] Note that when $a = 0$, the classification method resembles the nearest neighbour classification method, where the examples are used as a training set. On the other hand, when $a = 1$, the method is similar to naive Bayesian learning. However, in both cases the method represents a modification of the mentioned approaches, as the classes used in formula (1) are not all classes, but the ones learned by the GA. The experimental results presented in the following section we used 0.5 as the value of this parameter.

term needs to be compared to. Once learned, VSPs can readily be used during classification, requiring no processing during this phase. The search space constraining is very important for broad classification schemes. For example, UMLS has 135 classes and it would not be efficient to apply the nearest neighbour approach and compare a given term to all classes.

8 Experiments and Evaluation

In this section we describe the results of the classification experiments. First, we provide details on the experimental set-up and the evaluation framework. In order to evaluate our classification methods automatically, only terms classified in the ontology were used in the experiments. Namely, it can be checked in the ontology whether such terms were correctly classified by comparing the predicted classes to the classes assigned to them in the ontology. During the phase of retrieving the verb-term co-occurrences (see Section 4.3), some of the classified terms were singled out for testing, while the remaining terms were used for training (see Section 5). Namely, for each verb, approximately 10% of the retrieved terms were randomly selected for testing, and the union of all such terms formed a testing set (217 terms) for the classification task. The remaining terms constituted a training set (2392 terms) and were used for the learning of VSPs. Based on the training set, domain-specific verbs were associated with their VSPs (see Table 4 for examples). Then, each term from the training set was associated with the verb it most frequently co-occurred with. The VSP learnt for that verb was used to classify the term in question. Table 5 shows the results for some of the terms from the testing set and compares them to the correct classifications obtained from the ontology.

Note that in UMLS one term can be assigned to multiple classes. We regarded a testing term to be correctly classified if the predicted class was among these classes. Also, we accepted parents of correct classes as correct predictions. The reason for this is that such predictions are usually accepted when evaluation is performed manually. For example, `glucose` is classified in the ontology as a `carbohydrate` (a class subsumed by the class of `organic chemicals`), which would be a perfect prediction. However, the classification of `glucose` as an `organic chemical` would generally be accepted as a

correct classification, because `glucose` as a `carbohydrate` is also an `organic chemical`. The same evaluation strategy has been applied to the baseline method. We used a naive Bayes classifier whose goal is to maximise the conditional probability of a given term being assigned to a specific class based on the features used to represent the term. Each term is represented as a bag of co-occurring words, i.e. all single words occurring with the given term within a sentence. The aforementioned conditional probability is then estimated as the product of the class probability (estimated as the ratio between the number of all terms labelled with the given class and the total number of terms) and the conditional probabilities of features given the class (estimated as the ratio between the number of times a given single word co-occurs with terms from the given class and the number of all single words co-occurring with terms from the given class).

Verb	VSP – method 1	VSP – method 2
activate	Receptor Hormone Organic Chemical Enzyme Immunologic Factor Hazardous or Poisonous Substance Pharmacologic Substance	Pharmacologic Substance Receptor Enzyme Hormone Organic Chemical Amino Acid, Peptide, Protein
bind	Organic Chemical Hormone Receptor Enzyme Hazardous or Poisonous Substance Immunologic Factor Pharmacologic Substance	Pharmacologic Substance Receptor Enzyme Hormone Organic Chemical Nucleic Acid, Nucleoside, or Nucleotide Amino Acid, Peptide, Protein

Table 4. Learnt verb complementation patterns

Term	Verb	Suggested class – method 1	Suggested class – method 2	Correct classes
4 hydroxy-tamoxifen	bind	Organic chemical	Organic chemical	Organic chemical
benzoic acid	activate	Organic chemical	Pharmacologic substance	Organic chemical Pharmacologic substance
testoster-one	inhibit	Pharmacologic substance	Hormone	Steroid Pharmacologic substance Hormone

Table 5. Examples of the classification results

Table 6 provides information on the performance of our classification methods for each of the considered verbs separately and for the combined approach in which the verb most frequently co-occurring with a given term was used for its classification. For both classification methods, the combined approach provided considerably higher recall and a slight improvement in precision compared to average values obtained with the same method for each of the verbs separately. The classification precision did not tend to vary considerably, and was not affected by the recall values. The recall could be improved by taking into account more domain-specific verbs, while the improvement of precision depends on proper tuning of: (1) the module for learning the VSPs, and (2) the similarity measure used for the classification. Another possibility is to generalise the classification method by relying on domain-specific lexico-syntactic patterns instead of verbs. Such patterns would have higher discriminative power than verbs alone. Moreover, they could be acquired automatically. For instance, the CP-value method can be used for their extraction from a corpus [29].

The difference in the performance of our two classification is insignificant. Although the learning strategies differ significantly, this difference is largely neutralised by the use of a probability factor either explicitly or implicitly (see Section 7). Namely, the first method learns classes and their probabilities explicitly and uses them during the actual classification. These probabilities explicitly encode a bias towards certain classes during the classification phase. In the second method, this bias is encoded during the learning phase by not accepting classes causing a high number of false positives and false negatives, and these classes are never compared by the similarity measure. The baseline method also encodes both general class probabilities and conditional probabilities of a class given context words. We believe that this method has been outperformed in both cases, because verbs (including their learnt selectional preferences) in combination with the CLS similarity measure are better class discriminators than are individual words in combination with word-class probabilities.

The values for precision and recall provided in Table 6 refer to the classification methods themselves. If they were to be used for the automatic ontology update, then the success rate of such update would also depend on the performance of the term recognition method, as the classification module would operate on

its output. We used the C/NC-value method for ATR; still any other method may be used for this purpose. We have chosen the C/NC-value method because it is constantly improving and is currently performing around 72% recall and 98% precision [27].

Verb	Method 1			Method 2		
	Recall	**Precision**	**F-measure**	**Recall**	**Precision**	**F-measure**
activate	19.28	66.59	29.90	19.79	66.31	30.48
bind	29.30	66.53	40.68	27.98	66.72	39.43
block	5.17	62.98	9.56	9.19	61.42	15.99
inhibit	16.62	62.81	26.28	16.58	62.76	26.23
interact	13.16	64.31	21.85	13.40	63.89	22.15
mediate	11.68	62.75	19.69	10.17	63.02	17.51
modulate	10.44	64.13	17.96	10.47	64.08	18.00
regulate	4.83	61.08	8.95	5.73	62.58	10.50
repress	6.18	62.91	11.25	7.09	63.14	12.75
stimulate	9.39	63.25	16.35	12.24	62.63	20.48
Average:	12.61	63.73	20.25	13.26	63.66	21.35
Combined:	**49.92**	**64.17**	**56.16**	**52.78**	**63.84**	**57.79**
Baseline:	**42.81**	**55.32**	**48.27**	**42.81**	**55.32**	**48.27**

Table 6. The performance of the classification method

Currently, it is difficult to compare different term classification approaches reported in literature due to the lack of common evaluation for different systems and technologies [8, 33]. For example, the corpora used for testing differ significantly in content and size, terms are treated differently, classification focuses on different classes, errors are not treated uniformly, etc. A fair comparison of two term classification methods would require the usage of the same classification scheme and the same testing corpus. Many of alternative term classification approaches were either unavailable or designed for specific classes. Nonetheless, most of the results were reported for fewer number of classes, while the probability of missing a

correct class increases with the higher number of classes available. It is then natural for the performance measures to provide "poorer" values when tested on broader classification schemes. Also, methods targeted at specific classes are able to explore their characteristics in more detail and incorporate them manually into the method, which usually results in better performance. Still, they are limited to specific classes and cannot be applied in general case.

With these points in mind, let us discuss our approach in relation to some of the recently reported term classification in the biomedical domain. In Table 7, we compare the classification methods described earlier in Section 2 with respect to:

- the methodological approach taken,
- the types of features used,
- the number of target classes and dependence of the method on these classes,
- the classification performance evaluated by F-measure.

Most frequently, a ML approach is adopted as opposed to rule-based approaches. Still, even in the ML approaches, the utilised features are usually hand-crafted and targeted towards specific classes (e.g. methods described in [13-15]), which reduces the flexibility of such methods in the same way rules do (e.g. [9]). With that respect, our method belongs to the minority of ML classification approaches together with [12, 18, 19], which are able to infer class specific features rather than simply make use of manually extracted features.

Finally, our method achieved 58% for F-measure when tested on 28 classes, while the average value of F-measure is 72% for the average number of 7 classes. Bearing in mind that the probability of missing a correct class generally increases with the number of target classes (e.g. F-measure in [18] dropped from 84% to 73% when going from two to three classes), we can say that our method is in line with state-of-the-art methods for classification of biomedical terms.

Ref.	Approach	Features				Classes		F-measure
		internal	external	domain	ling.	#	fixed	
this	GA + NN		✓	✓	✓	28		58
[9]	rules	✓		✓		2	✓	97
[6]	rules	✓	✓	✓	✓	6	✓	83
[12]	SVM		✓			11		60
[13]	SVM	✓		✓		6	✓	54
[14]	SVM		✓	✓	✓	6	✓	67
[15]	SVM	✓		✓	✓	10	✓	74
[16]	naive Bayes	✓		✓		5	✓	66
[18]	naive Bayes		✓		✓	3		76
[19]	HMM		✓		✓	12		73

Table 7: A comparison of related approaches

9 Conclusion

Efficient update of the existing knowledge repositories in the rapidly expanding domain of biomedicine is a burning issue. Due to an enormous number of terms and the complex structure of the terminology, manual update approaches are prone to be both inefficient and inconsistent. Thus, it has become absolutely essential to implement efficient and reliable term recognition and term classification methods as means of maintaining the knowledge repositories. In this paper, we have suggested two term classification methods. For the preliminary experiments, we used the UMLS ontology in the domain of biomedicine, but the method can be easily adapted to use other ontologies.

The classification method makes use of the contextual information. Not all word types found in the context are of equal importance in the process of reasoning about the terms: the most informative are verbs, noun phrases (especially terms) and adjectives [4]. The presented term classification approach revolves around domain-specific verbs. These verbs are automatically identified and used to collect unclassified terms and to suggest their potential classes based on the automatically learnt verb complementation

patterns defined as collections of terms and classes that could be used in combination with domain-specific verbs.

We presented two machine learning methods, to obtain selectional preferences representing the hypotheses about the classes of terms used with the corresponding domain-specific verbs. The first method is a simple generalisation algorithm based on the partial order between classes induced by the ontology. The second method is a standard genetic algorithm designed to optimise verb selectional patterns in terms of their precision and recall. The first method also produces the estimation of the conditional probability of a class given the verb. This information is used to rank classes according to their probability. In the second method, all classes in a pattern are treated equally, i.e. there are no preferences towards specific classes in a pattern. However, the very presence of a class in a pattern suggests it as a highly probable class for the given verb.

Once the verb selectional patterns are available, they are used to propose classes for the terms co-occurring with domain-specific verbs. Note that not every term appearing in a corpus is guaranteed to be classified by the proposed classification method due to the fact that a term need not occur as a complement of a domain-specific verb. However, the generality of the method can be improved by applying the same approach to other types of linguistic or terminological elements. For example, terms from certain classes tend to co-occur with terms from a restricted set of classes (e.g. genes with proteins, hormones with receptors, etc.). Even more, the method can be further generalised to use a combination of lexical classes, which can be specified as a set of lexico-syntactic patterns. Further experiments with the generalisation of the classification methods by basing them on a set of domain-specific lexico-syntactic patterns instead of domain-specific verbs are expected to demonstrate better performance in terms of recall and precision.

References

[1] MEDLINE. 2003. National Library of Medicine: Available at: http://www.ncbi.nlm.nih.gov/PubMed/

[2] Altman R, Bada M, Chai X, Carillo M, Chen R. 1999. RiboWeb: An Ontology-Based System for Collaborative Molecular Biology. IEEE Intelligent Systems 14(5):68-76.

[3] Baker P, Goble C, Bechhofer S, Paton N, Stevens R, Brass A. 1999. An Ontology for Bioinformatics Applications. Bioinformatics 15(6):510-520.

[4] Maynard D, Ananiadou S. 2000. Identifying Terms by their Family and Friends. Proceedings of 18th International Conference on Computational Linguistics (COLING 2000), Universitat des Saarlandes, Saarbrucken, Germany, 530-536.

[5] UMLS. 2003. National Library of Medicine. Available at: http://www.ncbi.nlm.nih.gov/PubMed/

[6] Narayanaswamy M, Ravikumar K, Vijay-Shanker K. 2003. A Biological Named Entity Recognizer. Proceedings of the 8th Pacific Symposium on Biocomputing (PSB 2003), Lihue, Hawaii, USA, 427-438.

[7] Oliver D, Rubin D, Stuart J, Hewett M, Klein T, Altman R. 2002. Ontology Development for a Pharmacogenetics Knowledge Base. Proceedings of the 7th Pacific Symposium on Biocomputing (PSB 2002), Hawaii, USA, 65-76.

[8] Blaschke C, Valencia A. 2002. Molecular Biology Nomenclature Thwarts Information Extraction Progress. IEEE Intelligent Systems - Trends & Controversies 17(3):73-75.

[9] Fukuda K, Tsunoda T, Tamura A, Takagi T. 1998. Toward Information Extraction: Identifying Protein Names from Biological Papers. Proceedings of the 3rd Pacific Symposium on Biocomputing (PSB 1998), Hawaii, USA, 707-718.

[10] IUPAC. 2004. IUPAC Nomenclature of Organic Chemistry. Available at: http://www.acdlabs.com/iupac/nomenclature/

[11] Nedellec C. 2002. Bibliographical Information Extraction in Genomics. IEEE Intelligent Systems - Trends & Controversies 17(3):76–80.

[12] Stapley B, Kelley L, Sternberg M. 2002. Predicting the Sub-Cellular Location of Proteins from Text Using Support Vector Machines. Proceedings of the 7th Pacific Symposium on Biocomputing (PSB 2002), Hawaii, USA, 374–385.

[13] Kazama J, Makino T, Ohta Y, Tsujii J. 2002. Tuning Support Vector Machines for Biomedical Named Entity Recognition. Proceedings of the ACL Workshop on Natural Language Processing in the Biomedical Domain, Philadelphia, PA, USA, 1-8.

[14] Lee K, Hwang Y, Rim H. 2003. Two-Phase Biomedical NE Recognition based on SVMs. Proceedings of ACL Workshop on Natural Language Processing in Biomedicine, Sapporo, Japan, 33-40.

[15] Takeuchi K, Collier N. 2003. Bio-Medical Entity Extraction using Support Vector Machines. Proceedings of ACL Workshop on Natural Language Processing in Biomedicine, Sapporo, Japan, 57-64.

[16] Nobata C, Collier N, Tsujii J. 2000. Automatic Term Identification and Classification in Biology Texts. Proceedings of the Natural Language Pacific Rim Symposium (NLPRS 2000), Beijing, China.

[17] Mitchell T. 1997. Machine Learning. McGraw Hill, p. 414.

[18] Hatzivassiloglou V, Duboue P, Rzhetsky A. 2001. Disambiguating Proteins, Genes and RNA in Text: A Machine Learning Approach. Bioinformatics 1(1):97-106.

[19] Collier N, Nobata C, Tsujii J. 2001. Automatic Acquisition and Classification of Terminology Using a Tagged Corpus in the Molecular Biology Domain. Journal of Terminology, 239-257.

[20] Harris Z. 1968. Mathematical Structures of Language. Wiley.

[21] Hindle D. 1990. Noun Classification from Predicate-Argument Structures. Proceedings of 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh, Pennsylvania, USA, 268-275.

[22] Resnik P. 1992. A Class-Based Approach to Lexical Discovery. Proceedings of 30th Annual Meeting of the Association for Computational Linguistics, Newark, Deleware, USA, 327-329.

[23] Nenadic G, Spasic I, Ananiadou S. 2002. Automatic Discovery of Term Similarities Using Pattern Mining. Proceedings of 2nd International Workshop on Computational Terminology – CompuTerm 2002, Taipei, Taiwan, 43-49.

[24] Spasic I, Nenadic G, Manios K, Ananiadou S. 2002. Supervised Learning of Term Similarities. Intelligent Data Engineering and Automated Learning – IDEAL 2002, H. Yin, et al. (Eds), Springer-Verlag, 429-434.

[25] Frantzi K, Ananiadou S, Mima H. 2000. Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method. International Journal on Digital Libraries 3(2):115-130.

[26] Nakagawa H, Mori T. 1998. Nested Collocation and Compound Noun for Term Recognition. Proceedings of First Workshop on Computational Terminology (COMPUTERM 98), Montreal, Canada, 64-70

[27] Nenadic G, Spasic I, Ananiadou S. 2002. Automatic Acronym Acquisition and Management within Domain-Specific Texts. Proceedings of 3rd International Conference on Language, Resources and Evaluation, Las Palmas, Spain, 2155-2162.

[28] Goldberg D. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, p. 432.

[29] Nenadic G, Spasic I, Ananiadou S. 2004. Automatic Discovery of Term Similarities Using Pattern Mining. Terminology 10(1):55-80.

[30] Bodenreider O, Burgun A, Rindflesh T. 2002. Assessing the Consistency of a Biomedical Terminology through Lexical Knowledge. International Journal of Medical Informatics, 67(1-3):85-95.

[31] Grefenstette G, Hearst M. 1992. Method for Refining Automatically-Discovered Lexical Relations: Combining Weak Techniques for Stronger Results. Proceedings of AAAI Workshop Statistically-Based Natural Language Programming Techniques, Menlo Park, Canada.

[32] Hearst M. Automatic Acquisition of Hyponyms from Large Text Corpora. Proceedings of the 14th International Conference on Computational Linguistics (COLING-92), Nantes, France.

[33] Blaschke C, Hirschman L, Valencia A. 2002. Information Extraction in Molecular Biology. Briefings in Bioinformatics 3(2):154-165.